

RESEARCH OUTPUTS / RÉSULTATS DE RECHERCHE

Programmation tangible

Henry, Julie; Boraita, Fanny; Dumas, Bruno

Published in:

Des ressources numériques pour ressourcer les pratiques

Publication date:

2019

Document Version

le PDF de l'éditeur

[Link to publication](#)

Citation for pulished version (HARVARD):

Henry, J, Boraita, F & Dumas, B 2019, Programmation tangible: vers une évaluation des concepts perçus par l'enfant à travers la manipulation d'un outil. Dans *Des ressources numériques pour ressourcer les pratiques: Actes du 2e colloque scientifique Ludovia#CH*. p. 9-12, Ludovia#CH 2019, Yverdon-Les-Bains, Suisse, 16/04/19. <https://ludovia.ch/2019/wp-content/uploads/2019/04/Actes-2019_V4.pdf>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Programmation tangible : vers une évaluation des concepts perçus par l'enfant à travers la manipulation d'un outil

Boraita Fanny^{1,2}, Dumas Bruno¹, Henry Julie^{1,2}

¹Institut NADI, Faculté d'informatique, Université de Namur

²Institut IRDENA, Université de Namur

Résumé. Cette étude vise à répondre à la question suivante : est-ce que l'enfant a conscience des concepts-clé en programmation qu'il met en œuvre lorsqu'il utilise un outil tangible ? Une analyse de ces outils est au cœur de l'étude afin de pointer, entre autres, leurs finalités didactiques : la mise en œuvre des concepts et leur perception. Combinés à des observations d'activités avec des enfants de 3 à 10 ans, les résultats de l'analyse permettront de créer des ressources afin d'aider les enseignants à mesurer le développement des compétences en programmation de leurs élèves. Le premier résultat est l'élaboration d'une classification des outils tangibles.

Mots-clés. outil tangible, programmation, évaluation, compétences, concepts-clé.

1 Introduction

Déjà dans les années 80, les capacités en résolution de problèmes, raisonnement logique et analyse de formes géométriques dans leurs similitudes et leurs différences se révélaient être meilleures chez les enfants qui utilisaient le langage LOGO (Siegler, 2001). A l'heure actuelle, de plus en plus d'outils spécifiquement conçus pour développer précisément la pensée informatique sont utilisés dans les classes mais leur finalité didactique reste souvent dans l'ombre des *softskills*. En effet, si l'idée que l'initiation à la programmation est une aide au développement de compétences transversales est toujours partagée par les spécialistes en éducation, elle n'en reste pas moins difficile à prouver. Malgré l'abondance d'activités recourant au numérique, les enseignants manquent de ressources pour évaluer les compétences en programmation de leurs élèves. Notre étude vise à récolter des données afin de créer des outils pour aider cette évaluation. Tout d'abord, l'objectif de l'étude sera expliqué au regard de la littérature scientifique et du contexte actuel de l'enseignement en Fédération Wallonie-Bruxelles (FWB) dans lequel elle s'inscrit. Ensuite, la méthodologie sera présentée. Enfin, les premiers résultats seront exposés et discutés au regard des travaux futurs.

2 Contexte, ancrages théoriques et objectifs

L'interaction tangible est une approche privilégiée pour aider les enfants à développer des compétences en résolution de problèmes (Shaer et al., 2010). Les travaux de Papert (1980) et Perlman (1976) sont considérés comme pionniers dans l'apprentissage de la programmation à l'aide de robots ou de dispositifs embarqués. Cependant, la réponse apportée par le tangible au défi d'enseigner la programmation reste partiellement couverte. Si certains s'intéressent depuis peu à la compréhension "profonde" qu'ont les enfants des concepts-clé en programmation (Buffum et al., 2018 ; Hermans et al., 2018 ; Jin, 2018 ; Žanko, 2018 ; Grover & Basu, 2017 ; Rich et al., 2017) comme résultats observés suite à l'utilisation d'un outil d'aide à cet apprentissage (Horn et al., 2009 ; Xie et al., 2008), beaucoup continuent d'évoquer la simple performance (tâche réussie) (Sapounidis et al., 2015) ou des aspects liés à l'engagement, le plaisir, la motivation et la collaboration. De plus, dans la majorité des cas, un tel outil d'aide à l'apprentissage « *n'est pas choisi parce qu'il facilite l'apprentissage, mais plutôt parce qu'il est actuel (ou) parce qu'un supérieur le souhaite* » (Vandeput & Henry, 2018). Si jusqu'à présent la FWB avait opté pour une entrée en matière timide, en équipant des écoles de Thymio ou de makeblock via les projets École Numérique, l'enseignement de la programmation (à travers l'algorithmique) va être (ré)introduit dès la prochaine rentrée scolaire. La Belgique francophone va, elle aussi, posséder un référentiel officiel, inspiré du cadre européen de compétences numériques (DigComp¹). Sans remettre en doute les qualités de ces outils fournis, il est légitime de se questionner sur leur efficacité dans le cadre de ce nouvel enseignement de la programmation. Le Pacte pour un Enseignement d'Excellence² stipule que « *dès le primaire, une initiation à la logique du numérique peut utilement être réalisée par la programmation de machines simples* ». Il est par ailleurs indiqué l'existence prochaine d'un tronc commun polytechnique dans l'enseignement obligatoire (élèves de 5 à 15 ans) dans lequel les sciences informatiques prendront place : « *le numérique jouera un rôle essentiel au sein du domaine*

¹ <https://ec.europa.eu/jrc/en/digcomp>

² http://www.pactedexcellence.be/wp-content/uploads/2017/04/NOTE_GVT_PACTE_01_2015.pdf

d'apprentissage qui rassemble les mathématiques, les sciences, les compétences manuelles, techniques et technologiques, en sensibilisant progressivement aux sciences informatiques, notamment algorithmiques, dès le tronc commun ». Dans ce contexte où le numérique s'intègre de manière transversale, la question de la perception voire de l'assimilation des concepts sous-jacents aux sciences informatiques se pose. Encore trop peu de recherches se sont intéressées à la mesure du développement des compétences numériques chez les enfants, particulièrement à travers l'utilisation d'un outil tangible (Marinus et al., 2018 ; Bers, 2010). L'objectif de cette étude est d'apporter des éléments de réponse à cette question : est-ce que l'enfant a conscience des concepts de programmation qu'il met en œuvre lorsqu'il utilise un tel outil ? Par le biais d'une analyse des finalités didactiques des outils tangibles et l'observation d'activités de classe, il s'agira de rendre compte du développement de compétences en programmation³, à travers la perception des concepts sous-jacents (est-ce l'élève a conscience des concepts mis en œuvre ?) voire leur assimilation et acquisition (est-ce que l'élève a acquis les concepts ? est-ce qu'il les a assimilés ?).

3 Méthodologie

À partir de la littérature scientifique, l'analyse fine des outils tangibles a été entreprise sur leurs finalités didactiques, à savoir la mise en œuvre des concepts-clé en programmation et leur perception. Les résultats de l'analyse seront combinés avec l'observation d'activités auprès d'enfants de 3 à 10 ans. Elles seront conçues de manière à proposer une tâche complexe, essentielle au développement des compétences (Beckers, 2002). Les enfants manipuleront le Bee-bot, le Blue-bot ou le Ozobot. Ces outils tangibles ont notamment été choisis pour leur disponibilité commerciale et la richesse des ressources pédagogiques les accompagnant. Notre intérêt s'est également porté sur les différents designs d'interaction proposés à travers eux, à savoir respectivement une programmation via des boutons placés sur le robot, via une barre de programmation accompagnée de cartes (représentant les instructions possibles), ou encore via une application et son langage de programmation par blocs. Dans la lignée des travaux de Perrenoud (1997) : c'est en observant les élèves en activité que nous décortiquerons le développement de leurs compétences. Des interviews compléteront ces observations. À partir de propos, de ressentis, de situations précises, ces méthodes visent la qualité de la compréhension de l'information plutôt que son exactitude scientifique (Huberman, 1983). Ceci cadre clairement avec notre objet et notre public de recherche à savoir la perception qu'ont les jeunes enfants d'utiliser des concepts. L'observation des activités sera guidée par quatre étapes : 1) entretien avec l'enseignant afin d'identifier son *background* numérique ainsi que celui de sa classe ; 2) focus groupe avec les enfants quant à leurs perceptions préalables des concepts visés par l'activité ; 3) observation de l'activité et des interactions ; 4) focus groupe avec les enfants et les enseignants guidé par l'analyse des produits afin d'identifier le développement d'une compréhension des concepts-clé de programmation.

4 Résultats intermédiaires et discussion

Cette étude s'appuie sur l'analyse de la classification originale des outils tangibles de Henry (2018, voir Figure 1). Outre l'âge du public visé par chaque outil, cette classification se base d'une part, sur des aspects didactiques touchant les concepts-clé en programmation et d'autre part, sur des aspects propres aux outils eux-mêmes, notamment les techniques d'interaction mises en place à travers l'outil. Ainsi, au niveau didactique, une distinction est à faire entre les concepts-clé mis en œuvre par l'enfant lors de l'utilisation de l'outil et ceux qu'il perçoit réellement (sans aller jusqu'à dire « comprendre »). En ce qui concerne les concepts mis en œuvre, il est intéressant de constater que la programmation envisagée par les concepteurs d'outils tangibles privilégie la notion de séquence (voir Figure 1). Seuls six outils considèrent le concept de structure conditionnelle ; douze, le concept de boucle ; six, les variables et cinq, les sous-fonctions. Toutefois, la littérature scientifique n'apporte pas de réponse lorsqu'il s'agit de déterminer les concepts réellement perçus par l'enfant. L'étude décrite ici tend à remédier à ce manque en fournissant des outils d'évaluation.

Si les critères mis en évidence dans la classification de Henry (2018) complètent ceux proposés dans d'autres études, elle a le mérite de les enrichir. C'est notamment le cas pour la caractérisation physique de Yu et Roque (2018 - voir critères C5 et C6 dans la figure 1). Bien qu'intéressante, cette caractérisation mériterait une description plus poussée des blocs de programmation. On pourrait notamment souligner à travers ceux-ci les « contraintes syntaxiques » posées par le concepteur en vue d'aider l'enfant dans sa construction, ou encore la représentation

³ Inspirées des référentiels éprouvés (Berry, 2013) et du référentiel en cours de développement dans le Projet Erasmus+ "Pensée informatique et algorithmique dans le fondamental" de l'INRIA Lorraine, l'Université de Liège et l'Université de Saarland.

physique utilisée pour certains concepts-clé en programmation. De plus, le fait que Yu et Roque (2018) parlent de « donnée » plutôt que de variable nous pousse à envisager, pour la suite, de prendre en compte et de différencier (en termes de représentation) les rôles de la variable.

Figure 1. Classification des outils tangibles de Henry (2018)

	A						D											
	1	2	0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1
1 Algoblock																		
2 Bee-Bot *																		
3 Dialando																		
4 Dr. Wagon																		
5 E-Block																		
6 Electronic Blocks																		
7 KIBO																		
8 Pleo																		
9 Quetzal																		
10 Robo-Blocks																		
11 StarLoop																		
12 Strawbies																		
13 Tangicons																		
14 Tan-Pro Kit																		
15 T-Butterfly																		
16 Tern																		
17 T-Maze																		
18 Toque																		
19 T-ProRob																		
20 TurTan																		

0. Non décrit dans l'article

A. Âge

- De 2 à 6 ans
- De 7 à 12 ans

B. Concepts de programmation mis en oeuvre

- Algorithme/Programme (sans boucle infinie)
- Séquence
- Structure conditionnelle
- Boucle
- Variable
- Sous-programme (fonction, procédure)

C. Dispositif physique

- Tout-en-un
- Composé
- Continu
 - Discret
 - Capteurs embarqués
 - Caméra vidéo
 - Microphone
 - Autres capteurs (environnement)
- LED, écran, robot
 - Haut-parleurs
 - Dispositifs haptiques
- Physique, avec de l'électronique (tous les éléments tangibles)
- Hybride, avec des blocs de programmation tangibles

Dans la présente étude, d'autres outils tangibles, notamment commerciaux, ont augmenté le panel déjà analysé, à savoir l'Ozobot et le Blue-bot (voir figure 2). Une analyse de ces outils a mis en évidence la présence (ou l'absence) de certaines notions de programmation, outre les concepts de base qui nous intéressent plus particulièrement dans cette étude (Figure 2).

Figure 2. Caractérisation des outils tangibles choisis pour l'étude

Robots	Notions de programmation au regard du design d'interaction du robot					
	Les instructions disponibles (utilisées dans le cadre de cette recherche)	La séquence d'instructions programmée	L'exécution du programme	La boucle (répétition)	La variable	Le reset de la mémoire
Bee-Bot	Avancer, reculer, tourner vers la droite, tourner vers la gauche, s'arrêter (pause)	Non-visible	Lancée par l'enfant via un bouton (GO)	Non présente	Non présente	Commandée par l'enfant via un bouton (X)
Blue-Bot et sa barre de programmation	Avancer, reculer, tourner vers la droite, tourner vers la gauche, s'arrêter	Visible sur la « barre » Notion de début et de fin de programme	Lancée par l'enfant via un bouton	Représentée par la syntaxe [] (début et fin de la boucle) Nombre d'itérations à préciser (X2, X4, etc.)	Itérateur	Commandée par l'enfant via un bouton sur le robot (X) Impossible via la barre
Ozobot niveau 1 et son application	Avancer (de 1, 2, 5 ou 10 pas), reculer (de 1, 2, 5 ou 10 pas), tourner vers la droite (angles de 45, 90 ou 180°), tourner vers la gauche (angles de 45, 90 ou 180°)	Visible sur l'application	Lancée par l'enfant via l'application ET un bouton sur le robot	Non présente	Présente à travers la possibilité de voir varier le nombre de pas et l'angle de rotation (blocs différents)	Un nouveau programme écrase le précédent
Ozobot niveau 2 et son application	Avancer (de 1, 2, 5 ou 10 pas), reculer (de 1, 2, 5 ou 10 pas), tourner vers la droite (angles de 45, 90 ou 180°), tourner vers la gauche (angles de 45, 90 ou 180°)	Visible sur l'application	Lancée par l'enfant via l'application ET un bouton sur le robot	Boucle infinie et boucle permettant de répéter X fois	Les variables (pas, angles et itérateur) sont à fixer via l'application	Un nouveau programme écrase le précédent

Il apparaît ainsi qu'à travers un set d'instructions basiques, le Blue-bot et l'Ozobot offrent la possibilité de mettre en œuvre, à différents niveaux, les concepts de boucle et de variable dans deux de ses rôles : l'itérateur et le « most-recent » conteneur (Sajaniemi, 2005). Ils sont notamment perceptibles à travers la syntaxe employée. Si l'Ozobot, de par son interaction via une application, représente la boucle comme un bloc extensible capable d'englober d'autres blocs, le Blue-bot est limité par des cartes au volume fixe. Cette contrainte des interfaces tangibles « physicalisées » (dans notre étude, la barre de programmation et les cartes associées) se retrouve de la même

façon dans la représentation de la variable. Cet échantillon diversifié que nous offre ces trois robots va rendre possible l'évaluation (par comparaison) non seulement du choix syntaxique utilisé (et donc de sa représentation en terme de design), mais également de la faisabilité d'introduire à certains âges tel ou tel concept de programmation.

5 Conclusion et travaux futurs

Notre étude s'appuie sur les travaux de Henry (2018) qui, à travers sa classification d'outils tangibles, a montré que les finalités d'apprentissage de la programmation (notamment à travers la perception des concepts-clé) n'étaient pas prises en compte. En approfondissant les critères, en augmentant le panel d'outils et en portant un regard sur leur manipulation par des enfants de 3 à 10 ans, notre étude a pour but d'observer des activités de programmation tangible afin de récolter des données qui serviront à se rendre compte des concepts que les élèves perçoivent et à en évaluer leur compréhension. Ces éléments aideront à la création de ressources pour mesurer le développement des compétences en programmation. Trois robots ont été sélectionnés sur base de leur diversité en terme de design d'interaction : le Bee-Bot et sa programmation via boutons, le Blue-bot programmable via une barre physique et des cartes représentant les instructions et enfin, l'Ozobot possédant sa propre application et son langage en blocs. Tous trois sont des robots commerciaux qui jouissent d'une notoriété internationale et sont l'objet de nombreuses ressources pédagogiques.

Références

- Beckers, J. (2002). *Développer et évaluer des compétences à l'école : vers plus d'efficacité et d'équité*. Bruxelles : Labor Éducation.
- Berry, M. (Computing At School, 2013). *Computing in the national curriculum: A guide for primary teachers*.
- Bers, M. U. (2010). The TangibleK Robotics program: Applied computational thinking for young children. *Early Childhood Research & Practice*, 12(2), n2.
- Buffum, P. S., Ying, K. M., Zheng, X., Boyer, K. E., Wiebe, E. N., Mott, B. W. and Lester, J. C. (2018). Introducing the Computer Science Concept of Variables in Middle School Science Classrooms. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 906-911). ACM.
- Grover, S., and Basu, S. (2017). Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and boolean logic. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (pp. 267-272). ACM.
- Henry, J. Dumas, B. and Bodart, A. (2018). Programmation tangible pour les enfants : analyse de l'existant, classification et opportunités. AFIHM. 30eme conférence francophone sur l'interaction homme-machine, Oct 2018, Brest, France. IHM-2018, 9p, TeC-Travaux en Cours.
- Hermans, F., Swidan, A. and Smit, M.. (2018). Programming Misconceptions for School Students. In *Proceedings of the International Computing Education Research Conference: ICER '18*. ACM.
- Huberman, M. (1983). S'évaluer pour s'illusionner? Promesses et écueils de l'évaluation adaptative/interactive des innovations scolaires. *Cahier du Groupe des Chercheurs Romands en Pédagogie/Société Suisse pour la Recherche en Education*, 9, 1-38.
- Jin, K. H. (2018). A Loopy Encounter: Teaching Elementary Students the Concept of Loops. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education* (pp. 1099-1099). ACM.
- Marinus, E., Powell, Z., Thornton, R., McArthur, G., and Crain, S. (2018). Unravelling the Cognition of Coding in 3-to-6-year Olds: The development of an assessment tool and the relation between coding ability and cognitive compiling of syntax in natural language. In *Proceedings of the 2018 ACM Conference on International Computing Education Research* (pp. 133-141). ACM.
- Papert, S.. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- Perrenoud, P. (2011). *Construire des compétences dès l'école*. Paris : ESF Editeur (3e édition)
- Perlman, R. (1976). *Using computer technology to provide a creative learning environment for preschool children*.
- Rich, K. M., Strickland, C., Binkowski, T. A., Moran, C., and Franklin, D. (2017). K-8 Learning Trajectories Derived from Research Literature: Sequence, Repetition, Conditionals. In *Proceedings of the 2017 ACM Conference on International Computing Education Research* (pp. 182-190). ACM.
- Ryokai, K., Lee M. J. and Breitbart, J. M.. (2009). Children's storytelling and programming with robotic characters. In *Proceedings of the seventh ACM conference on Creativity and cognition*. ACM, 19-28.
- Sajaniemi, J. (2005). Roles of variables and learning to program. In *Proc. 3rd Panhellenic Conf. Didactics of Informatics, Jimoyiannis A (ed) University of Peloponnese, Korinthos, Greece*.
- Sapounidis, T., Demetriadis, S. and Stamelos, I.. (2015). Evaluating children performance with graphical and tangible robot programming tools. *Personal and Ubiquitous Computing* 19, 1 (2015), 225-237.
- Shaer, O., Hornecker, E. and others. (2010). Tangible user interfaces: past, present, and future directions. *Foundations and Trends® in Human-Computer Interaction* 3, 1-2 (2010), 4-137.
- Siegler, R. (2001). *Enfant et raisonnement. Le développement cognitif de l'enfant*. Bruxelles: De Boeck Université.